Funktionsweise des RSA-Verfahrens

CrypTool-Team

November 2010

Kryptografie – wozu?

- Das Verschlüsseln von Nachrichten hat in der Geschichte der Menschheit schon immer eine wichtige Rolle gespielt. In jedem Zeitalter gab es **Informationen**, die vor anderen **geschützt** werden mussten.

- Gerade in der heutigen Zeit, dem Zeitalter des Internet, ist es wichtig, Daten zu schützen.

Daten im Internet gelangen über Umwege zum Empfänger.

An jeder Zwischenstation können die Daten abgefangen, mitgelesen und sogar verändert werden.

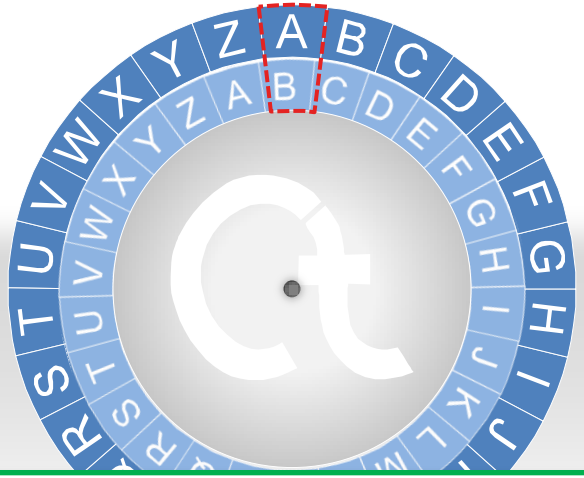
Moderne Kryptografie kümmert sich um den Schutz dieser Daten.

Einführendes Beispiel: Caesar-Verfahren

- Mit eines der ersten kryptologischen Verfahren war das Caesar-Verfahren.
  Der Name stammt vom römischen Kaiser Julius Caesar, der mit diesem Verfahren vor rund 2000 Jahren verschlüsselte Nachrichten an seine Generäle verschickte.
- Das Verfahren funktioniert wie folgt:

Klartext

Dies ist eine geheime Information!



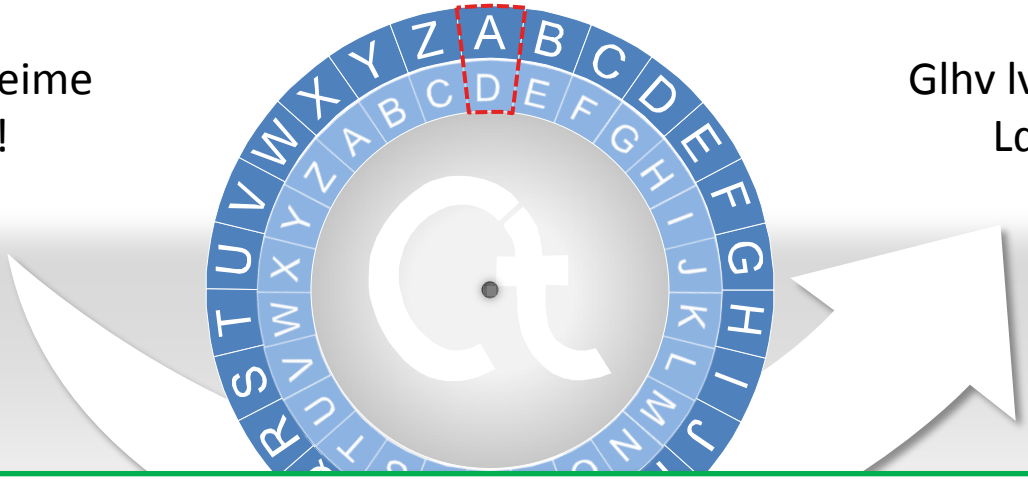
Das Alphabet wird zweimal untereinander geschrieben, allerdings versetzt, so dass A nicht unter A steht.

Einführendes Beispiel: Caesar-Verfahren

- Mit eines der ersten kryptologischen Verfahren war das Caesar-Verfahren.
  Der Name stammt vom römischen Kaiser Julius Caesar, der mit diesem Verfahren vor rund 2000 Jahren verschlüsselte Nachrichten an seine Generäle verschickte.
- Das Verfahren funktioniert wie folgt:

Klartext

Dies ist eine geheime Information!



Chiffrat

Glhv lvw hlqh jhkhlph Lqirupdwlrq!

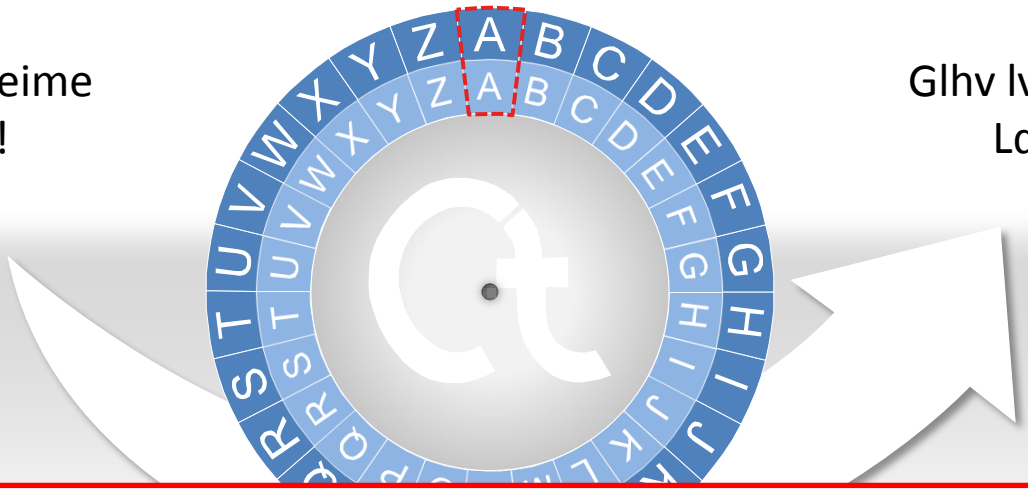
Nun wird jeder Buchstabe im Klartext durch den entsprechenden Buchstaben aus dem unteren Alphabet (der inneren Scheibe) ersetzt.

Einführendes Beispiel: Caesar-Verfahren

- Mit eines der ersten kryptologischen Verfahren war das Caesar-Verfahren.
  Der Name stammt vom römischen Kaiser Julius Caesar, der mit diesem Verfahren vor rund 2000 Jahren verschlüsselte Nachrichten an seine Generäle verschickte.
- Das Verfahren funktioniert wie folgt:

Klartext

Dies ist eine geheime Information!



Chiffrat

Glhv lvw hlqh jhkhlph Lqirupdwlrq!

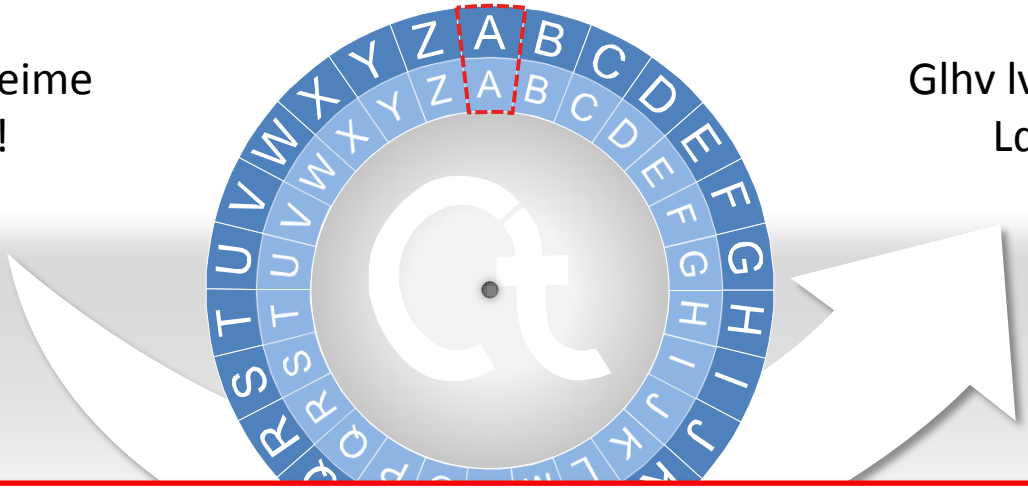
Es gibt nur 26 verschiedene Möglichkeiten, den Text zu verschlüsseln.
Dementsprechend lässt sich dieses Verfahren durch Ausprobieren leicht knacken.

Einführendes Beispiel: Caesar-Verfahren

- Mit eines der ersten kryptologischen Verfahren war das Caesar-Verfahren.
  Der Name stammt vom römischen Kaiser Julius Caesar, der mit diesem Verfahren vor rund 2000 Jahren verschlüsselte Nachrichten an seine Generäle verschickte.
- Das Verfahren funktioniert wie folgt:

Klartext

Dies ist eine geheime Information!



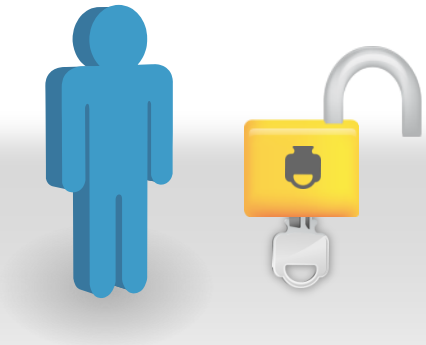
Chiffrat

Glhv lvw hlqh jhkhlph Lqirupdwlrq!

Möchten Sie das Verfahren an einem eigenen Text testen?
Ausprobieren lässt sich dies hier.

Gedankenmodell zum RSA-Verfahren

- Das Ziel ist es nun, eine Nachricht sicher zu verschicken.
  „Sicher“ bedeutet in diesem Fall, dass die verschlüsselte Nachricht möglicherweise **abgefangen** wird, aber dennoch **nicht gelesen** werden kann.
- Wie lässt sich dies realisieren? Eine zeitgemäße Antwort darauf liefert das **RSA-Verfahren**.
- Die Idee des Verfahrens lässt sich wie folgt veranschaulichen:



Jeder Teilnehmer hat ein Schloss mit passendem Schlüssel.

Gedankenmodell zum RSA-Verfahren

- Das Ziel ist es nun, eine Nachricht sicher zu verschicken.
  „Sicher“ bedeutet in diesem Fall, dass die verschlüsselte Nachricht möglicherweise **abgefangen** wird, aber dennoch **nicht gelesen** werden kann.
- Wie lässt sich dies realisieren? Eine zeitgemäße Antwort darauf liefert das **RSA-Verfahren**.
- Die Idee des Verfahrens lässt sich wie folgt veranschaulichen:

Die Idee ist, Schloss und Schlüssel voneinander zu trennen und Kopien des Schlosses zu veröffentlichen, den Schlüssel jedoch geheim zu halten.

Gedankenmodell zum RSA-Verfahren

- Das Ziel ist es nun, eine Nachricht sicher zu verschicken.
  „Sicher“ bedeutet in diesem Fall, dass die verschlüsselte Nachricht möglicherweise **abgefangen** wird, aber dennoch **nicht gelesen** werden kann.
- Wie lässt sich dies realisieren? Eine zeitgemäße Antwort darauf liefert das **RSA-Verfahren**.
- Die Idee des Verfahrens lässt sich wie folgt veranschaulichen:

Möchte man jetzt jemanden etwas schicken,
nimmt man dessen Schloss und verschließt die Nachricht damit.

Gedankenmodell zum RSA-Verfahren

- Das Ziel ist es nun, eine Nachricht sicher zu verschicken.
  „Sicher“ bedeutet in diesem Fall, dass die verschlüsselte Nachricht möglicherweise **abgefangen** wird, aber dennoch **nicht gelesen** werden kann.
- Wie lässt sich dies realisieren? Eine zeitgemäße Antwort darauf liefert das **RSA-Verfahren**.
- Die Idee des Verfahrens lässt sich wie folgt veranschaulichen:

Die Nachricht kann dann öffentlich verschickt werden, denn nur der richtige Empfänger kann mit dem passenden Schlüssel das Schloss wieder öffnen.

Zugrundeliegendes Problem

- Das RSA-Verfahren ist die elektronische Umsetzung des vorigen Gedankenmodells.
- Das Verfahren wurde nach seinen Erfindern **R**ivest, **S**hamir und **A**dleman benannt.
- Dem Verfahren liegt ein mathematisches Problem zugrunde. Im Fall von RSA ist es die Zerlegung in Primfaktoren. Dabei geht es darum, eine große Zahl als Produkt ihrer Primfaktoren darzustellen.
- Schwierig wird diese Zerlegung bei Zahlen, die nur aus großen Primfaktoren bestehen. Bisher gibt es kein effektives und schnelles Verfahren, um diese großen Primfaktoren zu erhalten. Und genau darauf beruht die Sicherheit des RSA-Verfahren. Dazu später mehr.

334780716989568987860441698482126908177047949837137685689124313889828837938780022876147116525317430877378144679994897

$$*$$

367460436667995904282446337996279526322791581643430876426760322838157396665112792333734171433968102700927987363089177

$$=$$

123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985690214341 3

Bit-Länge: 768 Dezimalstellen: 232

Aktuelle PCs können Zahlen mit etwa 80 Dezimalstellen schnell faktorisieren. Daher nutzt man RSA real mit Moduli von mindestens 300 Dezimalstellen.

Wie funktioniert das RSA-Verfahren

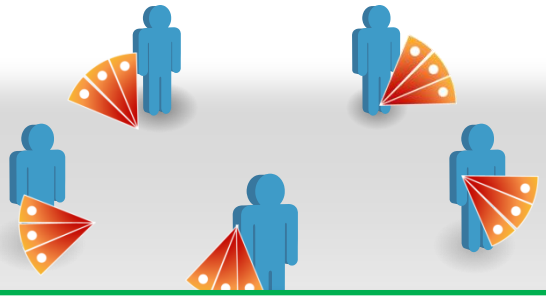
Um zu verstehen, wie das RSA-Verfahren funktioniert, benötigt man einige mathematische Grundlagen. Diese werden auf den nächsten Seiten erkärt.

1. Der Modulo-Operator
2. Eulersche φ- Funktion
3. Satz von Euler/Fermat

Mathematische Grundlagen - 1

Der Modulo-Operator

- Dieses Zeichen stellt den Modulo-Operator dar. Beim Modulo-Rechnen betrachtet man den Rest der ganzzahligen Division.
  D.h. man interessiert sich nur für den Rest, der beim Teilen entsteht, wenn man keine Nachkommastellen zulässt.
- Um es besser zu verstehen, können Sie sich folgendes Beispiel anschauen.



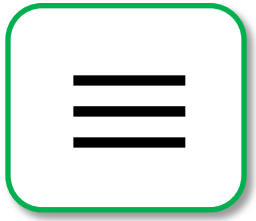
$$16 \equiv 1 \; mod \; 5$$

**Man möchte sich zu fünft einen Kuchen teilen, der in 16 Stücke aufgeteilt ist.
Jeder kann somit drei Stücke essen. Ein Stück bleibt übrig.
Genau diesen Rest berechnet der Modulo-Operator.**

Mathematische Grundlagen - 1

Der Modulo-Operator

- Dieses Zeichen stellt den Modulo-Operator dar. Beim Modulo-Rechnen betrachtet man den Rest der ganzzahligen Division.
  D.h. man interessiert sich nur für den Rest, der beim Teilen entsteht, wenn man keine Nachkommastellen zulässt.
- Um es besser zu verstehen, können Sie sich folgendes Beispiel anschauen.



Mathematische Definition

$$a \equiv b \; mod \; N$$

bedeutet, dass es eine ganze Zahl k gibt, so dass a sich darstellen lässt als

$$a = k * N + b$$

wobei für b gelten muss: $0 \leq b \leq N - 1$

Die Zahl k ist hierbei uninteressant. Wichtig ist nur, dass sie existiert.

Ein Beispiel für Modulo-Rechnen

Der Modulo-Operator ist mit den gewöhnlichen arithmetischen Operationen kommutierbar.
Konkret heißt dies, dass es egal ist, ob man **erst** z.B. eine Multiplikation ausführt,

oder aber erst modulo rechnet und **dann** multipliziert:

$$18 * 13 \equiv 8 * 3 \; mod \; 10$$
$$= 24 \; mod \; 10 \equiv 4 \; mod \; 10$$

Genauere Informationen können Sie im CrypTool-Skript finden (Kap. 4.4).

Mathematische Grundlagen - 2

Eulersche φ- Funktion

- Die eulersche φ - Funktion einer Zahl N gibt an, wie viele natürliche Zahlen es gibt, die kleiner als N und teilerfremd zu N sind.
- Als Formel sieht dies so aus:

$$\varphi(N) = \#\{a \in \mathbb{N} \mid ggT(a, N) = 1 \ und \ 1 \leq a < N\}$$?

Wichtige Eigenschaften der φ - Funktion

Für eine Zahl, die Produkt aus zwei Zahlen a und b ist, gilt:

$$\varphi(a * b) = \varphi(a) * \varphi(b)$$

Für Primzahlen p gilt:

$$\varphi(p) = p - 1$$

Für eine aus zwei Primzahlen zusammengesetzte Zahl $N = p * q$ gilt somit:

$$\varphi(N) = \varphi(p * q) = \varphi(p) * \varphi(q) = (p-1)(q-1)$$

Beispiel

Wir wollen $\varphi(10)$ berechnen:
Zunächst faktorisieren wir die Zahl

$$10 = 5 * 2$$

Da die Faktoren Primzahlen sind, können wir nun die Formel links verwenden:

$$\varphi(10) = \varphi(5) * \varphi(2) = 4 * 1 = 4$$

$$\varphi(5) = \#\{1, 2, 3, 4\} = 4 \qquad \varphi(2) = \#\{1\} = 1$$
$$\varphi(10) = \#\{1, 3, 7, 9\} = 4$$

Näheres siehe CrypTool-Skript, Kap 4.8.2

Mathematische Grundlagen - 3

Satz von Euler/Fermat

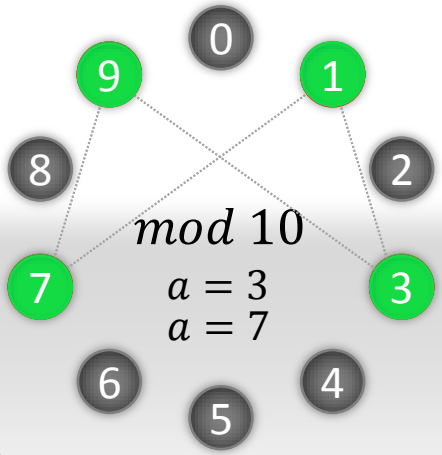
- Als letzte Grundlage benötigen wir noch den Satz von Euler/Fermat

Für zwei teilerfremde Zahlen a und N gilt:

$$a^{\varphi(N)} \equiv 1 \; mod \; N$$

Die Ergebnisse der Modulo-N-Operation sind immer Zahlen aus der **endlichen** Menge $\{0, 1, \ldots, N-1\}$. Funktionen nennt man **zyklisch**, wenn sich die Ergebnisse bei wiederholter Anwendung wiederholen.

Eine solche zyklische Funktion ist z.B. das Potenzieren mit fester Basis: Wir nehmen als Basis die Zahlen $a = 3$ und $a = 7$ und multiplizieren sie solange mit sich selbst, bis wir wieder bei der Zahl selbst landen. In unserem Beispiel ist $N = 10$ mit $\varphi(N) = 4$.



3 ▸ 9 ▸ 7 ▸ 1 ▸ 3

7 ▸ 9 ▸ 3 ▸ 1 ▸ 7

Die zwei Zyklen, die dabei entstehen, haben nun beide die Länge 4, was genau $\varphi(N)$ entspricht.

Multipliziert man eine solche Zahl a mit sich selbst, erreicht man in jedem Fall nach $\varphi(N)$ Multiplikationen wieder die Zahl a. Das sieht man, wenn man die obige Gleichung auf beiden Seiten mit a mulitpliziert.

Mit diesen Grundlagen können wir uns nun dem eigentlichen Verfahren zuwenden.

Schritt 1: Generieren der Schlüssel

- Das RSA-Verfahren wird auf den folgenden Folien in drei Schritten dargestellt.
- Als Erstes erzeugen wir uns nun ein RSA-Schlüsselpaar. Dieser Schritt ist nur einmal initial notwendig.

Formal

1. Wähle zwei Primzahlen p und q mit $p \neq q$
2. Bilde ihr Produkt $N = p * q$
3. Berechne den Wert der eulerschen φ - Funktion von N
$$\varphi(N) = \varphi(p * q) = (p-1)(q-1)$$
4. Wähle eine Zahl e, die zwischen 1 und $N-1$ liegt und teilerfremd zu $\varphi(N)$ ist.
5. Finde eine weitere Zahl d, für die gilt:
$$d * e \equiv 1 \; mod \; \varphi(N)$$

Am Beispiel

1. Wir wählen uns $p = 13$ und $q = 7$
2. Damit ist $N = 13 * 7 = 91$
3. $\varphi(91) = \varphi(13 * 7) = (13-1)(7-1) = 72$
4. Wir wählen $e = 5$, denn es gilt:
$ggT(5,72) = 1$
5. Wir nehmen $d = 29$, denn dann gilt:
$$d * e = 145 = 2 * 72 + 1 \equiv 1 \; mod \; 72$$

? Hier können Sie erfahren, wie sich eine solche Zahl d finden lässt. (erweiterter euklidischer Algorithmus)

(e, N) **ist der *öffentlichen RSA-Schlüssel.***
(d, N) **ist der *private RSA-Schlüssel.***

Näheres siehe CrypTool-Skript, Kap 4.10.3

Schritt 2: Verschlüsseln von Nachrichten

- Nun hat man die Voraussetzungen, um Nachrichten verschlüsselt zu versenden.
- Zuerst muss man die Buchstaben umwandeln, damit man mit ihnen auch rechnen kann.

Dazu kann man zum Beispiel folgende Ersetzung (Substitution) durchführen:

A	B	C	D	…	Z
01	02	03	04	…	26

Formal

Zum Verschlüsseln der Nachricht, muss nun

$$C = K^e \ mod \ N$$

gerechnet werden, wobei K der als Zahl codierte Klartext ist, und C die verschlüsselte Nachricht (das Chiffrat) darstellt. Die Zahlen e und N stammen aus dem öffentlichen RSA-Schlüssel (e, N).

Das hier dargestellte Verfahren ist deutlich vereinfacht.
Näheres auf den nachfolgenden Folien.

Beispiel

Wir führen unser Beispiel nun fort, wählen uns das Wort „GEHEIM“ und verschlüsseln es:

Buchstaben	G	E	H	E	I	M
in Zahlen	07	05	08	05	09	13

Nun wird G = 7 mit Hilfe der Formel links verschlüsselt.
Unser öffentlicher Schlüssel ist: $(5, 91)$

$$K^e = 7^5 = 7 * (7^2)^2 = 7 * (49)^2 \ \equiv 7 * 35 \ \equiv 63 \ mod \ 91$$

So wird „GEHEIM“ verschlüsselt in die Zahlen:

63	31	08	31	81	13

Schritt 3: Entschlüsseln von Nachrichten

- Der Empfänger erhält die verschlüsselte Nachricht.

Formal

Um die erhaltene, verschlüsselte Nachricht zu entschlüsseln, muss der Empfänger rechnen:

$$K = C^d \ mod \ N$$

Hierbei wird K den Klartext ergeben. Die Werte d und N entnimmt der Empfänger aus seinem privaten Schlüssel (d, N).

Beispiel

Die verschlüsselte Nachricht lautet:

63	31	08	31	81	13

Der Empfänger setzt in die Formel auf der linken Seite seinen geheimen Schlüssel $(29, 91)$ ein:

$$C^d = 63^{29} = \cdots \equiv 7 \ mod \ 91$$

Nach der Rechnung erhält er wieder den Klartext:

in Zahlen	07	05	08	05	09	13
Buchstaben	G	E	H	E	I	M

Wieso erhält man mit diesen Formeln am Ende wieder den Klartext?
Eine Erklärung folgt auf der nächsten Folie.

Was beim Ver- und Entschlüsseln geschieht

- Erklären lässt sich dies durch die Betrachtung folgender Formeln.

- Wir betrachten den Vorgang des Entschlüsselns der verschlüsselten Nachricht C genauer:

$$C^d = (K^e)^d = K^{e*d} \ mod\ N\text{, da } C = K^e \text{ (Verschlüsselung durch den Absender)}$$

- Es gilt $d * e \equiv 1\ mod\ \varphi(N)$, was man auch als $d * e = 1 + l * \varphi(N)$ auffassen kann, wobei l hier eine beliebige ganze Zahl ist.

- Damit gilt folgende Gleichungskette:

$$K^{e*d} = K^{1+l*\varphi(N)} = K * K^{l*\varphi(N)} = K * (K^{\varphi(N)})^l \ mod\ N$$

- Benutzt man nun den Satz von Euler/Fermat, $K^{\varphi(N)} \equiv 1\ mod\ N$, gilt:

$$K * (K^{\varphi(N)})^l \equiv K\ mod\ N$$

- Insgesamt folgt also, dass gilt:

$$C^d \equiv K\ mod\ N$$

Beim Potenzieren des Chiffrates mit dem privaten Schlüssel erhält man also wieder den Klartext.

Sicherheit des Verfahrens

- Das Verfahren auf der vorangehenden Folie wurde stark vereinfacht, um das Prinzip zu verdeutlichen. So, wie es dargestellt wurde, ist es noch nicht sicher.

G	E	H	E	I	M
07	05	08	05	09	13

63	31	08	31	81	13

.	E	.	E	.	.

Verschlüsselt man jeden Buchstaben einzeln, wird bei der Verschlüsselung jedem Buchstaben eindeutig eine bestimmte Zahl zugeordnet. Dies ist eine erste Angriffsmöglichkeit für eine **Häufigkeitsanalyse**, denn das Alphabet ist recht klein.
Bei einer solchen Analyse wertet man die Häufigkeiten der vorkommenden Werte aus und sortiert sie nach ihren Häufigkeiten. Dann kann man ausnutzen, dass Buchstaben in einer Sprache verschieden oft vorkommen. Im Deutschen ist der häufigste Buchstabe das „E“. Mit etwas Ausprobieren kann man so den Klartext erhalten.

- Um dieses Problem zu umgehen, fasst man mehrere Zahlen als Blöcke zusammen.
  In unserem Beispiel könnte man z.B. wie folgt zusammenfassen und dann erneut verschlüsseln:

GEH	EIM
070508	050913

Zu beachten ist hierbei, dass der Modulus N größer sein muss, als die maximale mögliche Zahl in einem Block.

- In der Praxis werden mit RSA keine Textblöcke verschlüsselt, sondern man kombiniert RSA mit einem symmetrischen Verschlüsselungsverfahren. Mit RSA wird nur dessen Schlüssel verschlüsselt (Hybridverschlüsselung).

Näheres siehe CrypTool-Skript, Kap 1.3

Zusammenhang Primfaktorzerlegung – RSA-Verfahren

- Wieso die Sicherheit des Verfahrens auf dem schwierigen mathematischen Problem der Primfaktorzerlegung beruht, wird nun geklärt.
- Dies lässt sich gut an unserem Beispiel erklären.
  Für unsere hier gewählte Zahl N lässt sich die Primfaktorzerlegung leicht finden:

$$N = 91 = 13 * 7 = p * q$$

- Somit lässt sich ebenfalls $\varphi(N)$ berechnen. Mit Hilfe des bekannten öffentlichen Schlüssels e und $\varphi(N)$ kann man nun den privaten Schlüssel d finden, da stets $d * e \equiv 1 \; mod \; \varphi(N)$ gelten muss. Hat man einmal den privaten Schlüssel gefunden, kann man die Nachricht entschlüsseln.
- Es hat noch niemand einen anderen Weg gefunden,
  - um aus dem öffentlichen Schlüssel das d zu berechnen, außer über die Primfaktorzerlegung.
  - um den Klartext aus dem Chiffrat ohne d zu berechnen.

Die Primfaktorzerlegung erlaubt also, aus dem öffentlichen Schlüssel (e, N) den privaten Schlüssel zu berechnen. Der Angreifer führt dabei nach der Primfaktorzerlegung nochmal den initialen Schritt 1 durch.

Weiterführende Links und Referenzen

- http://www.cryptool.org
  Ein OpenSource-Programm zum Erlernen und Lehren von kryptografischen Verfahren und zur Kryptoanalyse
- http://cryptool.org/download/CrypToolScript-de.pdf
  Skript zu CrypTool, das näher auf die Mathematik hinter den verschiedenen kryptografischen Verfahren eingeht
- https://www.datenschutzzentrum.de/selbstdatenschutz/internet/verschluesseln.htm
  Weiterführende Informationen zu Sicherheit und Datenschutz im Internet
- http://de.wikipedia.org/wiki/Kryptographie
  Allgemeiner Wikipedia-Artikel über Kryptografie
- http://www.xplora.org/downloads/Knoppix/MathePrisma/Start/
  Diverse mathematisch orientierte Workshops, unter anderem zu den Themen RSA und Caesar-Verschlüsselung
- http://de.wikipedia.org/wiki/RSA-Kryptosystem
  Wikipedia-Artikel über das RSA-Verfahren